Locator (URL) or other address directing the web browser to a particular server (such as server **912**), and the web browser may generate a Hyper Text Transfer Protocol (HTTP) request and communicate the HTTP request to the server. The server may accept the HTTP request and communicate to client system **920** one or more Hyper Text Markup Language (HTML) files responsive to the HTTP request. Client system **920** may render a webpage based on the HTML files from the server for presentation to the user. This disclosure contemplates any suitable webpage files. As an example and not by way of limitation, web pages may render from HTML files, Extensible HyperText Markup Language (XHTML) files, or Extensible Markup Language (XML) files, according to particular needs. Such pages may also execute scripts such as, for example, and without limitation, those written in JAVASCRIPT, JAVA, MICROSOFT SILVERLIGHT, combinations of markup language and scripts such as AJAX (Asynchronous JAVASCRIPT and XML), and the like. Herein, a reference to a webpage encompasses one or more corresponding webpage files (which a browser may use to render the webpage) and vice versa, where appropriate.

[0098] FIG. **10** is a functional diagram illustrating a programmed computer system in accordance with some embodiments. As will be apparent, other computer system architectures and configurations can be used to perform the described methods. Computer system **1000**, which includes various subsystems as described below, includes at least one microprocessor subsystem (also referred to as a processor or a central processing unit (CPU) **1006**). For example, processor **1006** can be implemented by a single-chip processor or by multiple processors. In some embodiments, processor **1006** is a general purpose digital processor that controls the operation of the computer system **1000**. In some embodiments, processor **1006** also includes one or more coprocessors or special purpose processors (e.g., a graphics processor, a network processor, etc.). Using instructions retrieved from memory **1007**, processor **1006** controls the reception and manipulation of input data received on an input device (e.g., image processing device **1003**, I/O device interface **1002**), and the output and display of data on output devices (e.g., display **1001**).

[0099] Processor **1006** is coupled bi-directionally with memory **1007**, which can include, for example, one or more random access memories (RAM) and/or one or more read-only memories (ROM). As is well known in the art, memory **1007** can be used as a general storage area, a temporary (e.g., scratch pad) memory, and/or a cache memory. Memory **1007** can also be used to store input data and processed data, as well as to store programming instructions and data, in the form of data objects and text objects, in addition to other data and instructions for processes operating on processor **1006**. Also as is well known in the art, memory **1007** typically includes basic operating instructions, program code, data, and objects used by the processor **1006** to perform its functions (e.g., programmed instructions). For example, memory **1007** can include any suitable computer-readable storage media described below, depending on whether, for example, data access needs to be bi-directional or uni-directional. For example, processor **1006** can also directly and very rapidly retrieve and store frequently needed data in a cache memory included in memory **1007**.

[0100] A removable mass storage device **1008** provides additional data storage capacity for the computer system **1000**, and is optionally coupled either bi-directionally (read/write) or uni-directionally (read-only) to processor **1006**. A fixed mass storage **1009** can also, for example, provide additional data storage capacity. For example, storage devices **1008** and/or **1009** can include computer-readable media such as magnetic tape, flash memory, PC-CARDS, portable mass storage devices such as hard drives (e.g., magnetic, optical, or solid state drives), holographic storage devices, and other storage devices. Mass storages **1008** and/or **1009** generally store additional programming instructions, data, and the like that typically are not in active use by the processor **1006**. It will be appreciated that the information retained within mass storages **1008** and **1009** can be incorporated, if needed, in standard fashion as part of memory **1007** (e.g., RAM) as virtual memory.

[0101] In addition to providing processor **1006** access to storage subsystems, bus **1010** can be used to provide access to other subsystems and devices as well. As shown, these can include a display **1001**, a network interface **1004**, an input/output (I/O) device interface **1002**, an image processing device **1003**, as well as other subsystems and devices. For example, image processing device **1003** can include a camera, a scanner, etc.; I/O device interface **1002** can include a device interface for interacting with a touchscreen (e.g., a capacitive touch sensitive screen that supports gesture interpretation), a microphone, a sound card, a speaker, a keyboard, a pointing device (e.g., a mouse, a stylus, a human finger), a global positioning system (GPS) receiver, a differential global positioning system (DGPS) receiver, an accelerometer, and/or any other appropriate device interface for interacting with system **1000**. Multiple I/O device interfaces can be used in conjunction with computer system **1000**. The I/O device interface can include general and customized interfaces that allow the processor **1006** to send and, more typically, receive data from other devices such as keyboards, pointing devices, microphones, touchscreens, transducer card readers, tape readers, voice or handwriting recognizers, biometrics readers, cameras, portable mass storage devices, and other computers.

[0102] The network interface **1004** allows processor **1006** to be coupled to another computer, computer network, or telecommunications network using a network connection as shown. For example, through the network interface **1004**, the processor **1006** can receive information (e.g., data objects or program instructions) from another network, or output information to another network in the course of performing method/process steps. Information, often represented as a sequence of instructions to be executed on a processor, can be received from and outputted to another network. An interface card or similar device and appropriate software implemented by (e.g., executed/performed on) processor **1006** can be used to connect the computer system **1000** to an external network and transfer data according to standard protocols. For example, various process embodiments disclosed herein can be executed on processor **1006** or can be performed across a network such as the Internet, intranet networks, or local area networks, in conjunction with a remote processor that shares a portion of the processing. Additional mass storage devices (not shown) can also be connected to processor **1006** through network interface **1004**.

[0103] In addition, various embodiments disclosed herein further relate to computer storage products with a computer-readable medium that includes program code for performing